



HAL
open science

A Fully Quantum Algorithm for Hydrodynamic Lattice Gas Cellular Automata

Niccolo Fonio, Giuseppe Di Molfetta, Pierre Sagaut

► **To cite this version:**

Niccolo Fonio, Giuseppe Di Molfetta, Pierre Sagaut. A Fully Quantum Algorithm for Hydrodynamic Lattice Gas Cellular Automata. 2024. hal-04543971

HAL Id: hal-04543971

<https://hal.science/hal-04543971>

Preprint submitted on 12 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Fully Quantum Algorithm for Hydrodynamic Lattice Gas Cellular Automata

Niccolo Fonio^{1,2*}, Giuseppe Di Molfetta¹ and Pierre Sagaut³

¹Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France.

²Università di Torino, Dipartimento di Fisica, Torino, Italia.

³Aix Marseille Univ, Centrale Marseille, CNRS, M2P2 UMR 7340, 38
rue Joliot-Curie, Marseille, 13013, France.

*Corresponding author(s). E-mail(s): niccolo.fonio@edu.unito.it;
Contributing authors: giuseppe.dimolfetta@lis-lab.fr;
pierre.sagaut@univ-amu.fr;

Abstract

Lattice Gas Cellular Automata (LGCA) are a computational model widely known and applied for the simulation of many physical phenomena. Their implementation requires an amount of resources and operations which scale linearly versus the system size and number of time steps. We propose a quantum-pointers-based quantum algorithm able to simulate LGCA while exhibiting an exponential advantage in space complexity and a number of quantum operations independent from the system size. We propose a collision circuit for the FHP lattice-gas automata considering the 2-, 3-, and 4-body collisions. These are implemented with two methodologies that suggest the procedure for finding quantum circuits for LGCA with more collisions. We also propose a phase estimation algorithm to retrieve information about a single cell, whose application can be expanded for implementing other collisions. A general methodology to identify the invariants associated to quantum LGCA is also proposed.

Keywords: Cellular Automata, Quantum Computing, Quantum Circuits, Quantum Advantage, Fluid-dynamic

1 Introduction

Since Computational Fluid Dynamics (CFD) algorithms for solving Navier-Stokes (NS) equations have been developed through the last decades depending on available technologies, the application of quantum computing is of major interest for fluid dynamic simulation [1], seeking a quantum advantage. For Quantum CFD there are, currently, several promising proposals that follow two main approaches.

The first one is based on computing the numerical solutions of NS equations, and it is addressed as the general field of solving linear or non-linear PDEs on a quantum computer. Several algorithms were proposed concerning this approach. This approach generally starts with the discretization of the PDE of interest and uses a quantum solver to find a solution. Concerning this possibility, a remarkable research [2] obtained an exponential advantage using the quantum amplitude estimation algorithm (QAEA) [3], that has been recently implemented in an efficient way [4], and has been also applied specifically to Burger's equation [5]. Quantum Walks have also been used as quantum numerical tools to simulate hydrodynamical equations [6, 7]. Other proposals regarding quantum numerical methods use a hybrid classical-quantum for accelerating parts of classical solvers or variational approach [8–11].

Alongside numerical methods for solving PDE, we find lattice methods, the aim of which is to simulate a fluid. Lattice Gas Cellular Automata (LGCA) and Lattice Boltzmann Methods (LBM) [12] belong to this class. LGCA considers a gas of particles propagating on a lattice with specific rules at a microscopic level, and its application for CFD started with 2D incompressible flows, with the landmark contributions of Frisch, Hasslacher, and Pomeau (FHP) [13] and Wolfram [14]. However, FHP for 2 dimensions and FCHC (Face Centered Hyper-Cube) [15–17] for 3 dimensions recovered NS equations with additional nonphysical spurious terms. The main issues were anomalous invariants, the lack of Galilean invariance, and the dependency of the pressure on the density. For FCHC these problems were overcome with a multi-speed model [18]. Regardless of more sophisticated models, the issues of LGCA moved the attention to LBM methods, which consider the propagation of probability density functions at the mesoscopic level according to the discrete velocity Boltzmann equation. Both LGCA and LBM were applied in several fields of application in physics, chemistry, biology [19–30]. The main technical issue of LGCA and LBM is scalability: large grids are often needed to simulate physical phenomena, and they quickly reach the limits of classical computers. In addition, LGCA and LBM modeling quantum phenomena may take advantage of quantum computers for many-body simulations. Scalability and many-body simulations are two of the reasons that motivated the formulation of Quantum LGCA [31–34]. However, the formulation proposed for Quantum LGCA does not provide a gate-based quantum algorithm, considering a stop-and-go procedure, with quantum collision and classical streaming, providing no advantage. It was experimentally proved with NMR [35], but has not been further developed. Up to now, to the best knowledge of the authors, no general quantum algorithm for LGCA has been proposed. For LBM, on the other side, there are promising proposals adopting Carleman linearization [36–38], and hybrid classical-quantum methods [39]. In particular, in [36], they consider unitary and non-unitary collision and streaming, comparing the space and computational complexity, without getting any advantage over classical

algorithms. The problems caused in the LBM approach by the non-linearity of the collision term and then the need for a linear embedding technique, and the necessity to stream PDF do not appear in LGCA. In LGCA, since we stream bits instead of PDF, the evolution in the case of detailed balance is linear: the non-linearity is given by the complex behavior of the whole system. This is the essential difference in the use of the encoding of the space regarding previous works. LGCA achieved good results for the simulation of several phenomena, despite issues with the initial formulation, overcome by more sophisticated models. Furthermore, LGCA with square lattice (HPP) and the FHP model (on the hexagonal lattice) we consider were found to be P-complete [40], thus predicting HPP or FHP-III (that considers specific collisions) is equivalent to calculating an arbitrary Boolean circuit. These reasons motivated the present work, whose aim is to present a quantum gate-based algorithm for the simulation on a quantum computer of LGCA, giving a specific example for the FHP model, since it can be considered as a starting point.

In this article we propose a quantum algorithm for simulating any detailed-balance LGCA getting a quantum exponential advantage in the number of qubits adopting the encoding of space proposed in [36] rephrased from a computational point of view with quantum pointers, and freeing the computational complexity from any dependency on the size of the system. This opens up the possibility of simulating LGCA with a number of cells never approachable with a classical computer. The perspectives for using this advantage will rely also on the capability of future research to model chemical, biological, and physical phenomena with LGCA. Further developments for LBM can be considered.

2 Results

In the following, we present our main results: 1) a quantum gate-based algorithm for LGCA simulation with unitary collision and streaming that provides an exponential advantage in space complexity, 2) a specific collision circuit for FHP in its first formulation, 3) a phase estimation algorithm to recover the single cell quantities such as mass and momentum and 4) considerations about quantum invariants of a quantum LGCA and a method to calculate their amount.

Quantum LGCA

We first consider that any LGCA with detailed balance can be implemented on a quantum computer. This is ensured by the Hermiticity of its transition matrix, from which we can formulate the unitary collision operator \hat{C} . This is possible assuming the standard Computational Basis Encoding (CBE) of the cell. CBE is explained in Methods section. The encoding of the lattice is performed with *quantum pointers*. For a DnQv model (with n the spatial dimension and v the number of discrete velocities, respectively), such pointers coincide with n superposed quantum registers for the position and one superposed quantum register for the velocity set. Then, using generalized Toffoli gates, each term of the superposition gets entangled, according to the presence of a particle. The entanglement allows us to perform the streaming among the pointers of the position and it also moves the particle that is pointed. As the first application,

let us consider the D2Q v model, where each node lives on a two-dimensional grid. In such situation, we consider three pointers : one for the row $|r\rangle$, one for the column $|c\rangle$, and a last one for the velocity $|v\rangle$. The presence of the particle is coded by one qubit $|n\rangle$ and we also need a supplementary cell register $|q\rangle$

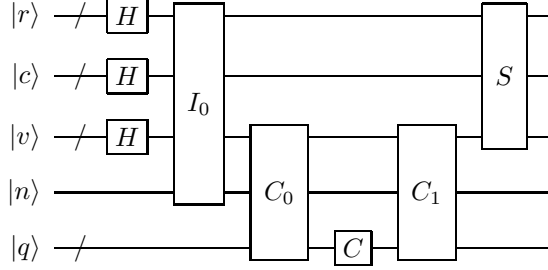


Fig. 1 Quantum gate-based circuit for simulation of LGCA on a quantum computer. I_0 is the initialization of the lattice made of generalized Toffoli gates. C_0 brings on a parallel ordered register of the occupation qubits: a computational basis encoding is adopted. C is the collision operator that depends on the LGCA to simulate. S is the streaming operator, characterized by a sequence of controlled adders that depend on the boundary conditions, number of dimensions, and number of quantum pointers

Pointers' registers undergo a series of Hadamard gates for creating a superposition of Nv indices, where N is the number of cells in the lattice and v is the number of velocities of the LGCA. We entangle the pointers to each particle in the lattice with \hat{I}_0 , which is performed by applying generalized Toffoli gates. Thus, we get the following state for the lattice

$$|\Psi\rangle = \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} |r\rangle |c\rangle \sum_{k=0}^{v-1} |k\rangle |n_k\rangle_{r,c} |q=0\rangle \quad (1)$$

Where $|\Psi\rangle$ is the state of the lattice, N_r and N_c are the number of rows and columns. Finally, we apply \hat{C}_0 , which brings all the occupation qubits in an ordered cell register $|q\rangle$.

$$|\Psi\rangle = \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} |r\rangle |c\rangle \sum_{k=0}^{v-1} |k\rangle |0\rangle |q_0 \dots q_v\rangle_{r,c} \quad (2)$$

This operation can be made with controlled swaps and a quantum adder for superposition states [41]. Now, after C_0 we apply the collision to the q register. The cell register is initialized with CBE, so the unitary collision deriving from the transition matrix previously mentioned is able to perform the collision step with \hat{C} . Then, we recreate the state of **Eq. 1** with the updated values of qubits. This operation, called C_1 takes up to $O(2^v)$ controlled operations. We conjecture that this cost can be lowered depending on the specific LGCA to simulate. This is a central feature of the algorithm to be made more efficient for the implementation of more precise models such as multi-speed models [18]. As a final step, we stream with \hat{S} , composed by a series of controlled adders previously mentioned. Each velocity streams differently. The explicit

form of each operation is provided in the discussion. This performs the mechanism of any LGCA with $O(\log_2(N)) + O(v)$ qubits and $O(I_0(N, v, T)) + O(2^v T)$ number of quantum gates. The cost of \hat{I}_0 depends on the lattice to be initialized, while the second term represents the cost of the evolution, which is independent of the system size. In fact, the collision and streaming operators act in parallel on each cell and on each pointer : that is the advantage of quantum parallelism. This leads to an exponential advantage in space and performs the evolution with a number of operations independent of the system size.

Quantum collision circuit

LGCA works with a discrete evolution separated into two steps: the collision step and the streaming step. The collision step rearranges the configuration of the cells: particles scatter within the site and change direction according to the conservation laws. The collision step can be seen as a rule that sends the state s to the state s' , where s' can be decided according to conservation laws to be applied. With the computational basis encoding, the collision step becomes an operator with the following effect

$$\hat{C} |s\rangle = |s'\rangle$$

Love [42] tried to figure out the operations to apply in order to get the desired collision, without implementing an overall circuit for the desired collision. As we show in **Fig.2**, we propose a series of operations that is able to perform 0-momentum collisions. The reasoning behind this collision algorithm is expanded in 3.3. It is based on a logic formulation of the 3-body collision and an equivalence class formulation for 2- and 4-body collisions. This collision subroutine takes advantage of quantum parallelism to lower the number of operations, which becomes independent of the system size. However, it is not possible to apply non-uniform random collisions, which is a fundamental aspect to avoid the classical model being chiral. The solution to the randomness of collision is left as a future perspective. The application of this collision operation is endorsed for the computational basis encoding. For applying it, is more straightforward to use pre-collision and post-collision operations involving the velocity register $|v\rangle$, the occupation register $|n\rangle$, and the cell register $|q\rangle$. While the pre-collision is composed of a series of controlled swaps and superposition quantum adder, the post-collision can be practiced in two ways. The first one is easier, and it considers reinitialization of the cell register with a complexity of $O(vt)$ where t is the number of time steps. Reinitialization can be avoided with additional qubits, which becomes $O(vt)$. This also allows us to store the state of the lattice at each time step, but increases the space complexity, conserving the advantage over the classical algorithm.

Quantum Phase Estimation

Regarding CBE, we propose a quantum phase estimation algorithm for retrieving general information about the cell without measuring it. We define, as in [42], quantum observables for the mass and momentum in x- and y-direction $\hat{m}, \hat{P}_x, \hat{P}_y$. These are generally sums of single-qubit gates, which are Hermitian. Thus, we can measure the

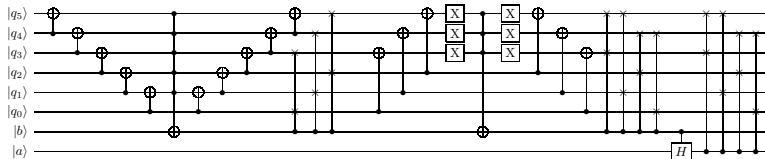


Fig. 2 Collisional circuit for 0-momentum collisions of FHP. These are performed depending on a conditional qubit $|b\rangle$ that gets flipped if the input state is a collisional state, and an additional qubit $|a\rangle$ that introduces the non-deterministic character of 2- and 4-body collisions. This circuit uses the respective invariances of collisional states.

mass or other observables of interest \hat{O} defining $\hat{U} = e^{2i\pi\hat{O}}$. This will turn the detection of those quantities of interest into a phase estimation problem.

Quantum invariants

The last result that opens some perspectives comes with the generalization of invariants. Classically a conserved quantity is a local quantity $O = f(\vec{x}, t)$ such that $f(\vec{x}, t) = f(\vec{x}, t+1)$, e.g. mass ($m = \sum_{i=0}^{v-1} m_i n_i(\vec{x}, t)$) and momentum ($\vec{p} = \sum_{i=0}^{v-1} \vec{c}_i n_i(\vec{x}, t)$). We can find a quantum counterpart of these quantities as proposed by Love [42]. This generalizes to the definition of a *quantum invariant*. A quantum invariant is an observable that commutes with the collision operator. Based on this definition, we find trivially, thanks to CBE, mass, and momentum as the sum of single quantum gates. We can calculate the number of quantum invariants by counting the number of independent solutions to a linear system, as explained in 3.5. This definition brings the number of quantum invariants to be higher than classical invariants. This is proved for D1Q3, where the quantum invariants were found to be 14. Anomalous classical invariants are a problem in the retrieval of NS behavior. The quantum invariants are not expected to be problematic since the evolution mimics the classical evolution. However, the physical reason behind these anomalous quantum invariants and their role in this quantum simulation of a classical system is left as a perspective for future research.

Perspectives

This article leaves several perspectives open. First, a way to characterize and optimize the initialization procedure is yet to be found. Since I_0 consists of a series of generalized Toffoli gates, it should be possible to rely on lowering the number of controls to initialize more cells with fewer control qubits. Symmetric or regular initialization could take advantage of quantum parallelism as well. Otherwise, an initial mechanism with collisions that do not preserve mass and momentum, followed by streaming, can be considered. In the second place, an improvement of \hat{C}_0 and \hat{C}_1 is necessary in order to get rid of the dependency on the number of velocities. In third place, future research should investigate the possibility of implementing a unitary collision operator for an LGCA with a semi-detailed balance. Another perspective is related to developing quantum circuits for the implementation of different collisions and different DnQv models. From this general implementation of the algorithm, it is possible to develop a

circuit for any DnQv model. The crucial question, which is a general problem of using a superposition, is how to retrieve the classical information. We conjecture that it should be possible to get global information about the system without perturbing the quantum state of the lattice. This will be possible with a deep modeling approach. The natural development of this algorithm may bring to the implementation of quantum LBM models, improving the existing proposals. Quantum circuits for different boundary conditions are yet to be found. The usage of the phase estimation algorithm can be expanded for the implementation of other collisions or retrieving information from the cells: it is possible to explore the global information about the system. The generalization of quantum invariants leaves open questions about their physical meaning and their possible role in these quantum simulations. As we said before, this algorithm makes it possible to simulate large grids, with an Avogadro's number of cells for example, using a relatively small amount of qubits, depending on the number of velocities to consider. With the current perspective on the technological development of quantum computers, simulations with this algorithm can take place in the future. The availability of large grids that quantum computing admits could flourish beginning with physical, chemical, and biological modeling with LGCA.

3 Methods

3.1 Introduction to FHP

A lattice gas cellular automata LGCA can generally be addressed as a DnQv model, where n is the space dimension and v is the number of velocities (bits) per cell, and it is characterized by the following elements [14] [12]

- A regular arrangement (lattice) of N cells in n dimensions
- Each cell has a discrete finite set of possible states represented by v bits
- Each cell state gets updated synchronously with collision and streaming operations

In this article we will address the FHP model, also called D2Q6 model, in two dimensions with 6 velocities on a hexagonal lattice, considering it as a first step toward more sophisticated models such as Digital physics [18]. The FHP set of velocities is the following.

$$\vec{c}_i = \left(\cos\left(\frac{\pi}{3}i\right), \sin\left(\frac{\pi}{3}i\right) \right) \text{ for } i = 0, 1, \dots, 5$$

Thus, each node of the lattice consists of a cell whose occupation state is defined by 6 bits $n_i(\vec{r}, t) \in \{0, 1\}$ for $i = 0, 1, \dots, 5$. $n_i(\vec{r}, t) = 1$ if a particle with velocity \vec{c}_i is present at node in position \vec{r} at time t , and $n_i(\vec{r}, t) = 0$ otherwise. The complete formulation of the model can be found in [12] [14]

Essentially, the LGCA can be computed as a grid of bits whose values evolve according to different rules. We will encode this grid of bits into a quantum system, performing the same evolution assured with Computational Basis Encoding. The dynamics consist of two steps: *collision* and *streaming*. An example of an evolution step is represented in **Fig.3**. The collision step modifies the state of each cell. For simulating NS equations, the fundamental constraint is to enforce elastic collisions, i.e.

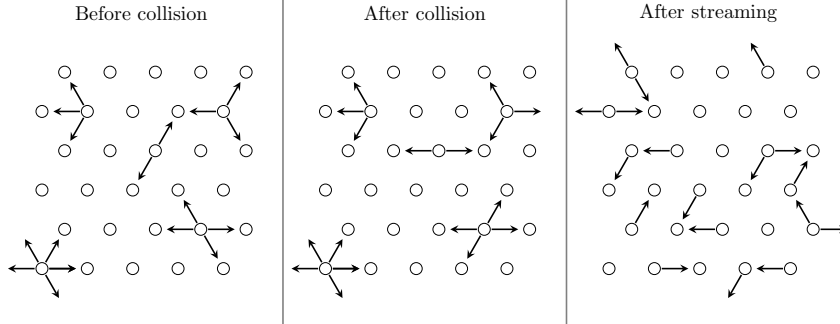


Fig. 3 Evolution step for FHP. Before collision is the starting state of the lattice. In the centre we see that the particles in cells with the collisional states of **Table 1** are rearranged. In the last part we see that particles have been streamed to neighbouring cells applying periodic boundary conditions

to introduce a collision operator such that the mass, i.e. $m = \sum_{i=0}^6 n_i$, and the total momentum of the cell, i.e. $\vec{p} = \sum_{i=0}^6 \vec{c}_i n_i$ and the kinetic energy $K = \sum_{i=0}^6 \vec{c}_i^2 n_i$ are conserved. This is needed for ensuring the conservation laws, that are treated with a Chapman-Enskog expansion, and retrieve at the second order NS-like equations. It is worth noting that on FHP's hexagonal lattice kinetic energy is conserved as soon as mass (i.e. the number of particles) is conserved since all discrete velocities have the same modulus. The conserved quantities in a classical simulation constrain the system to evolve preserving more quantities than needed. When we use a Quantum LGCA, we can define quantum invariants [42]. We show in 3.5 that quantum invariants of D1Q3 are 14 instead of 3, applying a generalized method. It is an open question to know if these quantum invariants have physical meaning. We conjecture they do not affect the simulation since the simulation of the classical system just depends on the collision, not explicitly on the conservation laws. Examples of possible collisions ensuring conservation of mass, momentum, and kinetic energy are given in **Table 1**

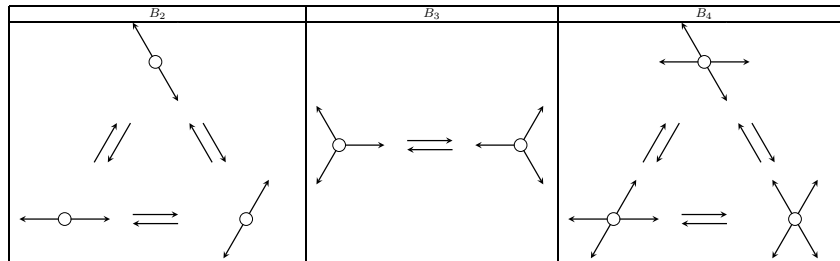


Table 1 Collisions of FHP. These are the 0-momentum collisions of the model

In general, as we said, any collision preserving mass (and then automatically kinetic energy) and momentum can be considered, because the form of the final equations does not depend on the collision. The transport coefficients, such as viscosity, generally depend on the collisions that we implement. The collisions given in **Table 1** are those that we are going to implement in a quantum circuit in 3.3.

We provide here the quantum version of an LGCA. First of all, we consider the *computational basis encoding* (CBE) for the bit-string of the cell, thus, any bit in the bit-string of a cell is turned into a qubit:

$$n_i \longrightarrow |n_i\rangle \forall i \in \{0, \dots, v\}$$

In the second place, we consider the transition matrix of an LGCA, which is defined as a square real matrix A with elements $a_{s,s'}$ such that $\sum_{s'} a_{s,s'} = 1$ and $a_{s,s'} \geq 0 \forall s, s'$. In particular, $a_{s,s'}$ is the probability that the cell state starts from s and evolves into s' . A deterministic transition matrix is a transition matrix where $a_{s,s'} = 1$ or $0, \forall s, s'$: there is not any randomness and each input state evolves into itself or into only one other state. A symmetric transition matrix is a transition matrix such that $a_{s,s'} = a_{s',s} \forall s, s'$, representing a detailed-balanced LGCA: the probability of $s \longrightarrow s'$ is the same probability that $s' \longrightarrow s$. FHP is a detailed-balanced LGCA. Since the transition matrix of a detailed-balanced LGCA is real and symmetric, it is trivially Hermitian. Since it is Hermitian, we can say that e^{iA} is unitary. Thus, we can simulate any detailed-balanced LGCA on a quantum computer using $\hat{C} = e^{iA}$ as the collision operator that acts on a quantum register given by CBE. If the LGCA is deterministic, the transition matrix corresponds to the matrix representation of the collision operator in the orthonormal basis provided by CBE. Thus, we can state the following property

Proposition 1. *The collision operator of any LGCA with detailed balance can be simulated on a quantum computer*

We can generally decompose any unitary operator in a quantum circuit with different methodologies, such as [43]. We will implement the collision for FHP with two methods based on the model itself. The methods to be preferred depend on the physical implementation of the quantum computer being used.

In general, an LGCA without detailed balance has a transition matrix that is not symmetric and thus is not Hermitian. In this case, it can be also possible to find a unitary able to simulate on a quantum computer the corresponding LGCA, but we leave it as an open perspective.

3.2 Quantum D2Q6

In this section, we give the specific formulation of the different operations previously mentioned that compose the overall circuit **Fig.1**. With quantum pointers encoding space and CBE, we have the following state for the lattice

$$|\Psi\rangle = \frac{1}{\sqrt{N2^{v'}}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} \sum_{k=0}^{2^{v'}-1} |r\rangle |c\rangle |k\rangle |n_k\rangle_i |q=0\rangle \quad (3)$$

Where $|r\rangle$ is the register of the row index, $|c\rangle$ is the register of the column index, $|k\rangle$ is the register of the velocity, N_r is the number of rows, N_c is the number of columns, $v' = \log_2(v)$ approximated by excess, $|n_k\rangle_{r,c}$ is the occupation qubit of the particle with velocity \vec{c}_k in position (r, c) , $|q\rangle$ is the cell-register, that must be in 0 state. Since the initialization procedure depends on the lattice, we give a general example for the following lattice

$$\begin{bmatrix} 010010 & 100001 \\ 000000 & 101010 \end{bmatrix}$$

With quantum pointers encoding of space and CBE the initial state of this lattice can be written as follows

$$\begin{aligned} |\Psi\rangle = \frac{1}{\sqrt{32}} [& |0\rangle |0\rangle (|0\rangle |0\rangle + |1\rangle |1\rangle + |2\rangle |0\rangle + |3\rangle |0\rangle + |4\rangle |1\rangle + |5\rangle |0\rangle) |q = 0\rangle + \\ & |0\rangle |1\rangle (|0\rangle |1\rangle + |1\rangle |0\rangle + |2\rangle |0\rangle + |3\rangle |0\rangle + |4\rangle |0\rangle + |5\rangle |1\rangle) |q = 0\rangle + \quad (4) \\ & |1\rangle |0\rangle (|0\rangle |0\rangle + |1\rangle |0\rangle + |2\rangle |0\rangle + |3\rangle |0\rangle + |4\rangle |0\rangle + |5\rangle |0\rangle) |q = 0\rangle + \\ & |1\rangle |1\rangle (|0\rangle |1\rangle + |1\rangle |0\rangle + |2\rangle |1\rangle + |3\rangle |0\rangle + |4\rangle |1\rangle + |5\rangle |0\rangle) |q = 0\rangle] \end{aligned}$$

Where $|i\rangle$ is intended to be the bit-representation of i for the velocity pointer.

The I_0 operation for this lattice will be the following one represented in **Fig.4**. Different combinations of controls over the same superposed registers make us access

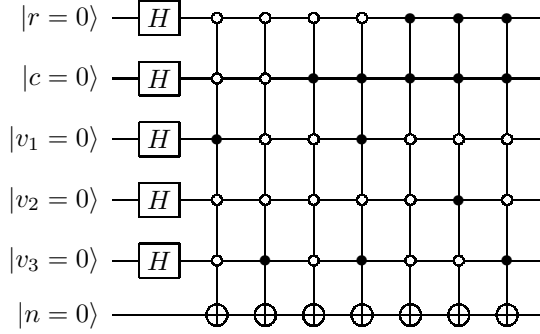


Fig. 4 Example of state preparation with quantum pointers

specific occupation qubits, that become entangled with the specific pointer. This is an example of massive initialization. As we said, a way to optimize this procedure is one of the open perspectives. After initializing the lattice, we act with C_0 , the pre-collision operation, which is an operation with the following action

$$C_0 \sum_{k=0}^{2^{v'}-1} |k\rangle |n_k\rangle |000000\rangle = \sum_{k=0}^{2^{v'}-1} |k\rangle |0\rangle |n_0 n_1 n_2 n_3 n_4 n_5\rangle$$

We first use a set of controlled-swaps to get the following state.

$$\sum_{k=0}^{v-1} |k\rangle |n = 0\rangle |n_k 2^k\rangle$$

Then, with the quantum adder of [41], we get the sum of all these superposed terms, describing the state of the cell in the computational basis encoding. Since the

quantum adder needs two copies of the same superposed state to be summed, as said before we need a parallel copy of the registers involved that are initialized identically. This is also caused by the no-cloning theorem. After the collision, the C_1 operation, called the post-collision operation, has to split the different particles giving them back to the respective velocity pointers. This can be done with specific operations depending on the number of velocities. For example, if we had two velocities we would need a circuit with the following effect

$$C_1 |q_0 q_1\rangle \longrightarrow \frac{|q_0 0\rangle |0\rangle + |0 q_1\rangle |1\rangle}{\sqrt{2}}$$

followed by the controlled swaps previously mentioned. This can be done with a quantum circuit using two additional qubits. The idea is to manipulate each possible configuration of the cell bringing it to the desired state. This procedure increases the number of operations, which becomes equivalent to the number of possible states of the cell. From here we have the exponential dependency on velocity $O(2^v)$ for each C_1 operation. The lowering of this cost is left as a future perspective. The streaming operator for FHP will be the one displayed in **Fig.5**. Moving a particle is interpreted as changing the corresponding pointer according to the velocity. In particular, referring to the grid, we will have the following analogies

$$\begin{aligned} \vec{c}_0 &\implies \text{rows} + 0, \text{cols} + 1 \\ \vec{c}_1 &\implies \text{rows} - 1, \text{cols} + 0 \\ \vec{c}_2 &\implies \text{rows} - 1, \text{cols} - 1 \\ \vec{c}_3 &\implies \text{rows} + 0, \text{cols} - 1 \end{aligned} \tag{5}$$

$$\begin{aligned} \vec{c}_4 &\implies \text{rows} + 1, \text{cols} + 0 \\ \vec{c}_5 &\implies \text{rows} + 1, \text{cols} + 1 \end{aligned} \tag{6}$$

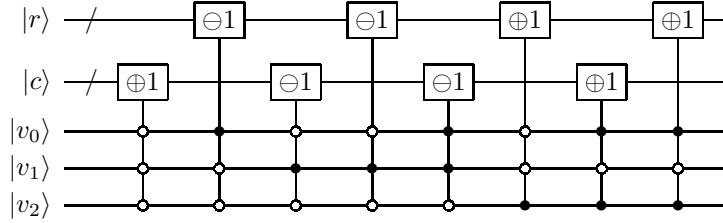


Fig. 5 Streaming step. The convention on the velocity index is the same as before. The adders are unitary operations such that the modulo ensures the periodic boundary condition

We see that we modify the pointers in space according to the velocity. The whole evolution of the lattice is represented as follows.

$$|\Psi_0\rangle = |0\dots 0\rangle |0\dots 0\rangle |000\rangle |0\rangle |000000\rangle \quad (7)$$

$$|\Psi_1\rangle = \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} \sum_{k=0}^{2^{v'}-1} |r(t)\rangle |c(t)\rangle |k\rangle |0\rangle |000000\rangle \quad (8)$$

$$|\Psi_2\rangle = \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} \sum_{k=0}^{2^{v'}-1} |r(t)\rangle |c(t)\rangle |k\rangle |n_k(t=0)\rangle_{r,c} |000000\rangle \quad (9)$$

$$|\Psi_3\rangle = \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} \sum_{k=0}^{2^{v'}-1} |r(t)\rangle |c(t)\rangle |k\rangle |0\rangle |n_0 n_1 n_2 n_3 n_4 n_5(t=0)\rangle_{r,c} \quad (10)$$

$$|\Psi_4\rangle = \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} \sum_{k=0}^{2^{v'}-1} |r(t)\rangle |c(t)\rangle |k\rangle |0\rangle |n_0 n_1 n_2 n_3 n_4 n_5(t=1)\rangle_{r,c} \quad (11)$$

$$|\Psi_5\rangle = \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} \sum_{k=0}^{2^{v'}-1} |r(t)\rangle |c(t)\rangle |k\rangle |n_k(t+1)\rangle_{r,c} |0\rangle \quad (12)$$

$$|\Psi_6\rangle = \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} \sum_{k=0}^{2^{v'}-1} |r(t) \oplus a_k\rangle |c(t) \oplus b_k\rangle |k\rangle |n_k(t+1)\rangle_{r,c} |0\rangle \quad (13)$$

$$= \frac{1}{\sqrt{Nv}} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} \sum_{k=0}^{2^{v'}-1} |r(t+1)\rangle |c(t+1)\rangle |k\rangle |n_k(t+1)\rangle_{r,c} |0\rangle \quad (14)$$

The state starts in $|\Psi_0\rangle$ with a series of $|0\rangle$ in **Eq.7**. We create the superposition of the pointers **Eq.8** with a series of Hadamard gates, getting $|\Psi_1\rangle$. We initialize the system entangling the presence qubit with the pointers through the application of generalized Toffoli gates **Eq.9**, getting $|\Psi_2\rangle$. An example of a circuit practicing this preparation is given in **Fig.4**. We bring the state of each cell on a parallel quantum register to perform the collision **Eq.10**, getting $|\Psi_3\rangle$. We perform the collision **Eq.11** bringing each cell to the updated state at $t+1$, getting $|\Psi_4\rangle$. We bring back the updated qubits to the values pointed by the row, column, and velocity pointers **Eq.12**, getting $|\Psi_5\rangle$. We perform the streaming with controlled adders updating the value of the row and the column with different values depending on the velocities **Eq.13**, getting $|\Psi_6\rangle$. In this step, the a_k and b_k are the ones previously mentioned for different velocities. For FHP this circuit is given in **Fig.5**. We can reorder the sum matching the row and column pointers and get the updated positions **Eq.14**. Then, we are ready to perform another time step, that will start back from **Eq.9**.

3.3 Collisional circuit for FHP

For performing the desired collisions in our algorithm, we start defining these in terms of quantum operations, interpreting them as rotations of the cell. Then, we propose an overall circuit for the implementation of 2-3-4-body collisions with 0-momentum,

represented in **Table 1**. We apply two different methodologies to find such a circuit. Each collision is implemented in two parts. The first part is the discrimination of the collisional states, and it uses two *conditional qubits*: $|b\rangle$ that is supposed to be $|1\rangle$ if we have a collisional input state, $|0\rangle$ otherwise, and $|a\rangle$ is used for creating the superposition for B_2 and B_4 .

As we can see from **Table 1**, collision B_3 can be interpreted as a rotation of 60° or 180° . Collisions B_2 and B_4 correspond to different rotations. We consider B_2 and B_4 to be implemented by one rotation of 120° and a successive rotation of 120° taking place with a probability of 0.5, resembling the model of FHP-I [12]. We point out that B_3 collisional states are invariant under a rotation of 120° , and B_2 and B_4 collisional states are invariant under a rotation of 180° . This will allow us to merge the circuit in a unique overall circuit for the three collisions. In terms of quantum operations, we can see rotations of the cell as a series of swaps operations, as shown in **Fig. 6**

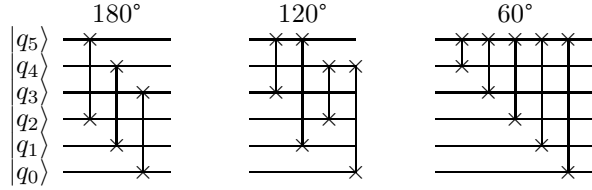


Fig. 6 Rotations with quantum circuits

To implement the discrimination of collisional states of B_3 we consider a logic methodology, starting from the expression that can be found in [12], adapted to our convention.

$$b = (q_0 \wedge q_1) \& (q_1 \wedge q_2) \& (q_2 \wedge q_3) \& (q_3 \wedge q_4) \& (q_4 \wedge q_5)$$

Where \wedge is a XOR operation and $\&$ is an AND operation. We can turn this logic expression in a quantum circuit interpreting \wedge as CNOT and $\&$ as a generalized Toffoli gate with the cell's qubits as controls and $|b\rangle$ as a target. Then we restore the original cell and we apply the C-Swap gates for a rotation of 180° with $|b\rangle$ as a control qubit. The circuit implementing B_3 is showed in **Fig. 7**

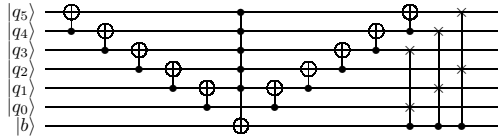


Fig. 7 B_3 collision

This circuit alters only the collisional states of B_3 , thus we will have the following evolution

$$\begin{aligned} |101010\rangle |00\rangle &\longrightarrow |010101\rangle |10\rangle \\ |010101\rangle |00\rangle &\longrightarrow |101010\rangle |10\rangle \end{aligned}$$

To implement B_2 and B_4 , we apply a reasoning based on equivalence classes. We define an *asymmetric opposite pair* as a pair of opposite bit-velocities that differ from each other (e.g. $|100000\rangle$ has 1 opposite pairs for the asymmetry between $|q_0\rangle$ and $|q_3\rangle$, $|110000\rangle$ has 2 opposite pairs for $|q_0\rangle|q_3\rangle$ and $|q_1\rangle|q_4\rangle$, $|100100\rangle$ has 0 opposite pairs). We consider the equivalence class of states with 0 asymmetric pairs.

As we can see, the collisional states of B_2 and B_4 belong to this class, and the other states are invariant under these collisions. Thus, we can target $|b\rangle$ in order to be $|1\rangle$ if there are no asymmetric pairs. Then with a series of C-Swaps with $|b\rangle$ as a control, we apply the first 120° rotation. Then, to recall the non-deterministic character of the rule, we use $|a\rangle$ to create a superposition between different outcomes. We apply a controlled-H gate with $|b\rangle$ as control and $|a\rangle$, initialized to $|0\rangle$, as target. Then we apply the same controlled rotation of 120° with $|a\rangle$ as a control.

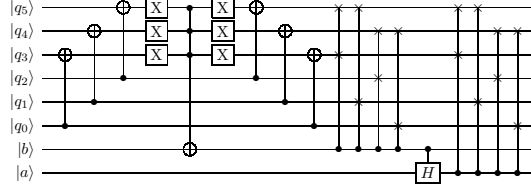


Fig. 8 $B_{2,4}$ (Head-on) collision

The circuit in **Fig.8** alters the collisional states of B_2 and B_4 , thus we will have the following evolution

$$\begin{aligned} |100100\rangle &\longrightarrow \frac{|001001\rangle |10\rangle + |010010\rangle |11\rangle}{\sqrt{2}} \\ |010010\rangle &\longrightarrow \frac{|100100\rangle |10\rangle + |001001\rangle |11\rangle}{\sqrt{2}} \\ |001001\rangle &\longrightarrow \frac{|010010\rangle |10\rangle + |100100\rangle |11\rangle}{\sqrt{2}} \\ |110110\rangle &\longrightarrow \frac{|101101\rangle |10\rangle + |011011\rangle |11\rangle}{\sqrt{2}} \\ |101101\rangle &\longrightarrow \frac{|011011\rangle |10\rangle + |110110\rangle |11\rangle}{\sqrt{2}} \end{aligned}$$

$$|011011\rangle \longrightarrow \frac{|110110\rangle |10\rangle + |101101\rangle |11\rangle}{\sqrt{2}}$$

As a natural consequence, these non-deterministic rules create entangled qubits within the cell. Superposition and entanglement are not physical features of a fluid dynamic simulation. Thus, a measurement of $|a\rangle$ can make the states collapse, exactly like a random extraction that takes place classically. However, there is a bias of such an implementation of non-deterministic rules: all the rotations will take place in the same way, so at each time step all the collisional states will have a rotation of the same angle, that can be different at the next time step. This represents a problem with the chirality of the models to be simulated. The consequences of this effect need to be explored in further studies. To apply this procedure we then need to reinitialize $|b\rangle$ and $|a\rangle$. This can be made by reapplying the discrimination of the collisional state part. These two circuits can be merged together for the invariances previously mentioned, as shown in **Fig.2** and their correct behavior has been verified on Quiskit.

We improved the circuits presented in [42], obtaining an overall circuit for the implementation of 0-momentum collisions. Moreover, we gave two possible methodologies to implement a collision, which can be used for other collisions. The formulation of other collisions is not trivial and requires a specific definition of equivalence classes or logic rules. The effect of avoiding the measurement of $|a\rangle$, propagating entangled qubits, is yet to be studied since the previous calculation of the evolution of the system holds if the cell can be represented as a product state. Thus, models of LGCA where there is a collision that creates entangled particles within the cell must be the subject of future studies.

3.4 Phase estimation algorithm

Staying within the cell, we can calculate different quantities of interest, such as momentum in the x and y direction and mass. In a QLGCA this retrieving of classical information has majorly been performed with a measurement and re-initialization of the cell [31] [39]. We show that with a phase estimation algorithm, we can avoid this stop-and-go procedure.

In [42] some specific operators whose eigenvalues can be interpreted as the respective quantities are defined. We can use these definitions since they depend on the CBE applied.

$$\hat{P}_x = Z_0 - Z_3 + \frac{Z_1 + Z_5}{2} - \frac{Z_2 + Z_4}{2} \quad (15)$$

$$\hat{P}_y = \frac{\sqrt{3}}{2}(Z_1 + Z_2 - Z_4 - Z_5) \quad (16)$$

$$\hat{m} = Z_0 + Z_1 + Z_2 + Z_3 + Z_4 + Z_5 \quad (17)$$

In general, these are not unitary operators, so we cannot apply them directly to the quantum register of a cell. However, we can apply complex exponentiation to get unitary operators $U_{\hat{O}} = e^{-2\pi i \hat{O}}$. In this way detecting the corresponding eigenvalue, thus the corresponding quantity of interest, becomes a phase estimation problem. With

this procedure, we take advantage of the fact that all the possible states of the cell are eigenstates of the mentioned operators since we are using a CBE. In the appendix, a specific example of momentum in x-direction can be found. In general, it is clear that for the CBE, the Z_i operator and their combination detect the state of the cell, which can be collisional. Thus, in an alternative scheme for the discrimination of collisional states, the conditional qubit $|b\rangle$ in **Fig.2** can be influenced by a phase estimation procedure for detecting a collisional configuration. For example, the operator $\hat{B}_3 = Z_0 - Z_1 + Z_2 - Z_3 + Z_4 - Z_5$ will have eigenvalue 0 if and only if there is a collisional state of B_3 . Thus, the PE procedure together with the interpretation of Z_i operators as operators that detect the presence of the i -th particle, due to the CBE adopted, can be considered another general procedure to practice a particular collision. The application of the phase estimation procedure here proposed does not provide a retrieval of classical information with quantum pointers' encoding. In fact, all the information about the cell is entangled to a superposed term of the quantum pointer register: this makes the retrieving of information particularly hard and should be investigated in future research.

3.5 Quantum invariants

In this section, we explain some fundamental features of this model arising from the implementation of LGCA on a quantum computer. This has a principal consequence in the conservation of local quantities [42] such as mass and momentum. Classically, for the discretization of the space, some unphysical invariants arise [44]. The purpose of many studies has been to get rid of them for getting better results. Moolvig and Texeira made extensive work [18] [45] for FCHC, which models a fluid in 3 dimensions.

Quantumly, we are dealing with observables. A physical interpretation of conserved quantities for a unitary evolution has been given [46] [47]. However, in this article, we address the computation of LGCA on a quantum computer, we do not propose a study of a quantum lattice gas. Thus, we will rely on the local conserved quantities defined by the commutation with the collision operator, as explained in the following. The conservation of these observables is the main topic of causal cellular automata, well defined in other works [48]. We define an observable \hat{O} as a quantum invariant if it commutes with the collision operator \hat{C} , so if it satisfies $[\hat{C}, \hat{O}] = 0$. We rewrite this property expanding the commutator as follows

$$\hat{C}\hat{O}\hat{C}^\dagger = \hat{O}' = \hat{O} \quad (18)$$

We can say that *any* operator \hat{O} respecting **Eq.18** is a quantum invariant. If we want to know how many quantum invariants there are for a QLGCA we apply the following theorem.

Theorem 2. *Consider a collision of a quantum $DnQv$ model as a unitary operator \hat{C} . Given a set of Pauli gates on the product space of the v qubits involved and their linear combination, the number of quantum conserved quantities of a $DnQv$ model is defined by the rank of the evolution matrix M , with elements*

$$M_{i,j} = \alpha_{i,j} - \delta_{i,j}$$

where

$$\hat{C}^\dagger \hat{O}_i \hat{C} = \sum_{j=0}^{4^v-1} \alpha_{i,j} \hat{O}_j$$

This is proved by the fact that the rank of the evolution matrix correspond to the number of independent solution of the linear system where each equation is the conservation equation for the combination of Pauli gates on v qubits. To give an example, the D1Q3 collision operator presented in [42] can be written in the following way

$$\hat{C}_{tot} = \frac{1}{4}(3III + IZZ + XXX + XYY - YXY + YYX - ZIZ + ZZI) \quad (19)$$

The previous procedure gives a rank for the evolution matrix equal to 14. This is the number of quantum invariants. Three of them are mass, momentum, and the identity. These are caused by the CBE applied, since they just "count" two quantities that are conserved by construction. The meaning of all the others has to be the subject of future studies. Thus, the formulation of a collision operator and the intuitive definition of a quantum invariant may hide several more invariants than mass and momentum for any quantum LGCA. However, the evolution of the system simulates the classical model. Thus, is probably true that the presence of multiple quantum conserved quantities does not affect the simulation model. This is true as long as this algorithm is used to simulate a classical LGCA. The quantum conserved quantities may affect the modeling of LGCA for other systems. Thus, the study of these conservation laws needs to be explored in more detail.

4 Conclusion

We proved that any detailed-balanced LGCA can be simulated with a quantum computer. A generalization to any LGCA, including semi-detailed balance, should be investigated in the future. Adopting the computational basis encoding for the cell state and a quantum pointers' encoding for the space allowed us to find an exponential advantage in space complexity. This can bring large-scale simulations using a relatively small number of qubits, accessible on quantum computers in the future, and with the possibility to characterize the interaction of particles within the cell in a classical or quantum way. The perspectives about simulating large-scale LGCA should be in the future to discover which phenomena can be modeled with an LGCA procedure. The advantage in computational complexity depends on the initialization: with massive initialization, needed to access each single particle, we do not get a strong advantage. Applying an alternative initialization, to be investigated in future works, brings to a cost of the initialization independent of the system size. A specific categorization of initialization procedures, also for ordered, semi-ordered, or random initial states, should be studied in the future. For the post-collisional operation, a quantum circuit without an exponential dependency on the number of velocities is to be found. For the streaming step of our algorithm using quantum pointers, we took inspiration from the classical reasoning of the algorithm, thus, we think that this can inspire also correct

streaming in the presence of different boundary conditions or obstacles. We presented a collisional circuit for the implementation of FHP collisions with 0 momentum. The proposed circuit is built up with two different methodologies for the discrimination of collisional states: the first one is based on logic, and the second one is based on equivalence classes of collisional states. These two methodologies can be applied to look for implementing new and different collisions. General considerations about the usage of a phase estimation algorithm have been proposed. This can be used for different purposes, including to implement an arbitrary collision, and to retrieve information about the cell on an additional register. This procedure does not admit to retrieving information from the quantum system with a quantum pointers' encoding of the space, for which a general procedure is yet to be found. We gave a general methodology for counting the quantum conserved quantities of a QLGCA, applying it to the case of the quantum version of D1Q3, finding more invariants than expected. While the conservation of mass and momentum can be deduced from the CBE encoding, the meaning of the other quantum-conserved quantities is yet to be found.

Acknowledgments. This work is supported by the PEPR integrated project EPiQ ANR-22-PETQ-0007, by the ANR JCJC DisQC ANR-22-CE47-0002-01 founded from the French National Research Agency and with the support of the french government under the France 2030 investment plan, as part of the Initiative d'Excellence d'Aix-Marseille Université - A*MIDEX AMX-21-RID-011.

Authors' contributions. NF designed the circuit-based model supervised by GDM and PS, and was a major contributor in writing the manuscript. All authors contributed to write, read and approved the final manuscript.

Data availability. Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Appendix A Phase estimation example

We provide a specific example for momentum in x-direction. The operator defined by Love, with the present convention is **Eq.15** The eigenvalues of this operator are $0, \pm 1, \pm 2, \pm 3$. We slightly modify P_x in order to get natural eigenvalues. Thus we define

$$\hat{P}_x = Z_0 - Z_3 + \frac{Z_1 + Z_1}{2} - \frac{Z_2 + Z_4}{2} + 3I \quad (\text{A1})$$

Thus, we get the eigenvalues to be $0, 1, 2, 3, 4, 5, 6$. In this way we have integer eigenvalues, and their bit-representation is precise with 3 qubits in the register for the phase estimation algorithm. We define $\hat{U}_x = e^{-2\pi i \frac{\hat{P}_x}{8}}$. We add a quantum register of 3 qubits where we store the eigenvalue, and we implement the circuit **Fig.A1**

We get $|p_{x_2} p_{x_1} p_{x_0}\rangle$ on the additional register, that is the bit representation of the momentum. Thus, we can measure the additional register in order to get the desired quantity. If $|u\rangle$ is a superposition of two eigenstates of \hat{U} , as happens after B_2 or B_4 , then we get on the additional register the superposition of the corresponding eigenvalues.

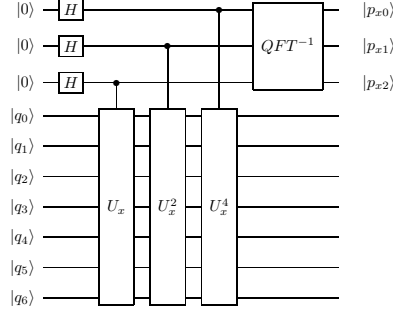


Fig. A1 Phase estimation algorithm for detecting momentum in x-direction

$$\begin{aligned}
 PE(\hat{U}) |000\rangle |u_1\rangle &= |\phi_1\rangle |u_1\rangle \\
 PE(\hat{U}) |000\rangle |u_2\rangle &= |\phi_2\rangle |u_2\rangle \\
 PE(\hat{U}) |000\rangle \frac{|u_1\rangle + |u_2\rangle}{\sqrt{2}} &= \frac{|\phi_1\rangle |u_1\rangle + |\phi_2\rangle |u_2\rangle}{\sqrt{2}}
 \end{aligned}$$

Thus, a measurement on the additional register would cancel any effect of superposition or entanglement that we can have in the cell. The phase estimation algorithm, merged with the interpretation of the operators given by Love [42], can be applied also using operators representing specific configuration on the cells.

An appendix contains supplementary information that is not an essential part of the text itself but which may be helpful in providing a more comprehensive understanding of the research problem or it is information that is too cumbersome to be included in the body of the paper.

References

- [1] Bharadwaj, S.S., Sreenivasan, K.R.: Quantum computation of fluid dynamics. arXiv preprint arXiv:2007.09147 (2020)
- [2] Gaitan, F.: Finding flows of a navier–stokes fluid through quantum computing. npj Quantum Information **6**(1), 61 (2020)
- [3] Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics **305**, 53–74 (2002)
- [4] Oz, F., San, O., Kara, K.: An efficient quantum partial differential equation solver with chebyshev points. Scientific Reports **13**(1), 7767 (2023)
- [5] Oz, F., Vuppala, R.K., Kara, K., Gaitan, F.: Solving burgers’ equation with quantum computing. Quantum Information Processing **21**, 1–13 (2022)
- [6] Hatifi, M., Di Molfetta, G., Debbasch, F., Brachet, M.: Quantum walk hydrodynamics. Scientific reports **9**(1), 2989 (2019)

- [7] Zylberman, J., Di Molfetta, G., Brachet, M., Loureiro, N.F., Debbasch, F.: Quantum simulations of hydrodynamics via the madelung transformation. *Physical Review A* **106**(3), 032408 (2022)
- [8] Steijl, R., Barakos, G.N.: Parallel evaluation of quantum algorithms for computational fluid dynamics. *Computers & Fluids* **173**, 22–28 (2018)
- [9] Demirdjian, R., Gunlycke, D., Reynolds, C.A., Doyle, J.D., Tafur, S.: Variational quantum solutions to the advection–diffusion equation for applications in fluid dynamics. *Quantum Information Processing* **21**(9), 322 (2022)
- [10] Chen, Z.-Y., Xue, C., Chen, S.-M., Lu, B.-H., Wu, Y.-C., Ding, J.-C., Huang, S.-H., Guo, G.-P.: Quantum approach to accelerate finite volume method on steady computational fluid dynamics problems. *Quantum Information Processing* **21**(4), 137 (2022)
- [11] Leong, F.Y., Ewe, W.-B., Koh, D.E.: Variational quantum evolution equation solver. *Scientific Reports* **12**(1), 10817 (2022)
- [12] Wolf-Gladrow, D.A.: *Lattice-gas Cellular Automata and Lattice Boltzmann Models: an Introduction*. Springer, ??? (2004)
- [13] Frisch, U., Hasslacher, B., Pomeau, Y.: Lattice-gas automata for the navier-stokes equation. *Physical review letters* **56**(14), 1505 (1986)
- [14] Wolfram, S.: Cellular automaton fluids 1: Basic theory. *Journal of statistical physics* **45**, 471–526 (1986)
- [15] Higuera, F.J., Jiménez, J.: Boltzmann approach to lattice gas simulations. *Europhysics letters* **9**(7), 663 (1989)
- [16] Rivet, J.-P., Henon, M., Frisch, U., d’Humières, D.: Simulating fully three-dimensional external flow by lattice gas methods. *Europhysics Letters* **7**(3), 231 (1988)
- [17] d’Humières, D., Lallemand, P., Frisch, U.: Lattice gas models for 3d hydrodynamics. *Europhysics Letters* **2**(4), 291 (1986)
- [18] Molvig, K., Donis, P., Miller, R., Myczkowski, J., Vichniac, G.: Multi-species lattice-gas automata for realistic fluid dynamics. In: *Cellular Automata and Modeling of Complex Physical Systems: Proceedings of the Winter School, Les Houches, France, February 21–28, 1989*, pp. 206–231 (1989). Springer
- [19] Mainster, M.A.: Cellular automata: retinal cells, circulation and patterns. *Eye* **6**(4), 420–427 (1992)
- [20] F Brandner, A., Timr, S., Melchionna, S., Derreumaux, P., Baaden, M., Sterpone, F.: Modelling lipid systems in fluid with lattice boltzmann molecular dynamics

- simulations and hydrodynamics. *Scientific reports* **9**(1), 1–14 (2019)
- [21] McCullough, J., Coveney, P.: An efficient, localised approach for the simulation of elastic blood vessels using the lattice boltzmann method. *Scientific Reports* **11**(1), 24260 (2021)
- [22] Nava-Sedeño, J.M., Hatzikirou, H., Klages, R., Deutsch, A.: Cellular automaton models for time-correlated random walks: derivation and analysis. *Scientific reports* **7**(1), 16952 (2017)
- [23] Roy, I., Ray, P.K., Balasubramanian, G.: Examining oxidation in β -nial and β -nial+ hf alloys by stochastic cellular automata simulations. *npj Materials Degradation* **5**(1), 55 (2021)
- [24] Stockman, H., Stockmant, C., Carrigan, C.: Modelling viscous segregation in immiscible fluids using lattice-gas automata. *Nature* **348**, 523–525 (1990)
- [25] Appert, C., Rothman, D.H., Zaleski, S.: A liquid-gas model on a lattice. *Physica D: Nonlinear Phenomena* **47**(1-2), 85–96 (1991)
- [26] Chopard, B., Droz, M.: Cellular automata model for heat conduction in a fluid. *Physics Letters A* **126**(8-9), 476–480 (1988)
- [27] Frenkel, D., Ernst, M.: Simulation of diffusion in a two-dimensional lattice-gas cellular automaton: a test of mode-coupling theory. *Physical review letters* **63**(20), 2165 (1989)
- [28] Simons, N.R., Bridges, G., Podaima, B., Sebak, A.: Cellular automata as an environment for simulating electromagnetic phenomena. *IEEE microwave and guided wave letters* **4**(7), 247–249 (1994)
- [29] Chen, S., Chen, H., Martnez, D., Matthaeus, W.: Lattice boltzmann model for simulation of magnetohydrodynamics. *Physical Review Letters* **67**(27), 3776 (1991)
- [30] Boghosian, B.M., Levermore, C.D.: A cellular automaton for burgers’ equation. *Complex Systems* **1**(1), 17–30 (1987)
- [31] Yepez, J.: Quantum lattice-gas model for computational fluid dynamics. *Physical Review E* **63**(4), 046702 (2001)
- [32] Yepez, J.: Quantum lattice-gas model for the diffusion equation. *International Journal of Modern Physics C* **12**(09), 1285–1303 (2001)
- [33] Yepez, J.: Quantum lattice-gas model for the burgers equation. *Journal of Statistical Physics* **107**, 203–224 (2002)
- [34] Yepez, J., Boghosian, B.: An efficient and accurate quantum lattice-gas model for

- the many-body schrödinger wave equation. *Computer Physics Communications* **146**(3), 280–294 (2002)
- [35] Pravia, M.A., Chen, Z., Yezpez, J., Cory, D.G.: Experimental demonstration of quantum lattice gas computation. *Quantum Information Processing* **2**, 97–116 (2003)
- [36] Itani, W., Sreenivasan, K.R., Succi, S.: Quantum algorithm for lattice boltzmann (qalb) simulation of incompressible fluids with a nonlinear collision term. arXiv preprint arXiv:2304.05915 (2023)
- [37] Todorova, B.N., Steijl, R.: Quantum algorithm for the collisionless boltzmann equation. *Journal of Computational Physics* **409**, 109347 (2020)
- [38] Itani, W., Succi, S.: Analysis of carleman linearization of lattice boltzmann. *Fluids* **7**(1), 24 (2022)
- [39] Budinski, L.: Quantum algorithm for the advection–diffusion equation simulated with the lattice boltzmann method. *Quantum Information Processing* **20**(2), 57 (2021)
- [40] Moore, C., Nordahl, M.G.: Predicting lattice gases is p-complete. Technical report, Citeseer (1997)
- [41] Lu, X., Jiang, N., Hu, H., Ji, Z.: Quantum adder for superposition states. *International Journal of Theoretical Physics* **57**, 2575–2584 (2018)
- [42] Love, P.: On quantum extensions of hydrodynamic lattice gas automata. *Condensed Matter* **4**(2), 48 (2019)
- [43] Krol, A.M., Sarkar, A., Ashraf, I., Al-Ars, Z., Bertels, K.: Efficient decomposition of unitary matrices in quantum circuit compilers. *Applied Sciences* **12**(2), 759 (2022)
- [44] Bernardin, D.: Global invariants and equilibrium states in lattice gases. *Journal of statistical physics* **68**, 457–495 (1992)
- [45] Teixeira, C.M.: Continuum limit of lattice gas fluid dynamics. (1994)
- [46] Meyer, D.A.: Quantum lattice gases and their invariants. *International Journal of Modern Physics C* **8**(04), 717–735 (1997)
- [47] Meyer, D.A.: Quantum mechanics of lattice gas automata: One-particle plane waves and potentials. *Physical Review E* **55**(5), 5261 (1997)
- [48] Schumacher, B., Werner, R.F.: Reversible quantum cellular automata. arXiv preprint quant-ph/0405174 (2004)